Visualization of Scalar Topology for Structural Enhancement

C. L. Bajaj and V. Pascucci
Department of Computer Sciences and TICAM
University of Texas, Austin

and

D. R. Schikore
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory

Visualization of Scalar Topology for Structural Enhancement

C. L. Bajaj   V. Pascucci
Department of Computer Sciences and TICAM
University of Texas, Austin, TX 78733

D. R. Schikore
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory, Livermore, CA 94550

Abstract

Scalar fields arise in every scientific application. Existing scalar visualization techniques require that the user infer the global scalar structure from what is frequently an insufficient display of information. We present a visualization technique which numerically detects the structure at all scales, removing from the user the responsibility of extracting information implicit in the data, and presenting the structure explicitly for analysis. We further demonstrate how scalar topology detection proves useful for correct visualization and image processing applications such as image co-registration, isocontouring, and mesh compression.

Keywords: Scientific Visualization, Scalar Fields, Curves and Surfaces, Vector Topology

1 Introduction

Visualization of scalar fields is common across all scientific disciplines, including geographic data such as altitude and temperature, medical applications with CT and MRI values, and pressure and vorticity magnitude in computational fluid dynamics. The purpose of the visualization is to aid the user in understanding the structure of the data[29].

Common methods for visualizing scalar fields can be grouped into two broad classes. First are methods whose aim is to detect structure and present a display to the user which communicates this structure. Critical to these methods is the definition of structure, and how well the definition matches the visualization users' need. Second are those methods which attempt to display the entire scalar field simultaneously, leaving interpretation of the display to the user. Combinations of the two methods serve to reinforce the information provided by each visualization. We will use for comparison one technique from each of these categories, isocontouring and colormapping.

Isocontours, or constant valued curves and surfaces from continuous 2D and 3D scalar fields, are a common visualization technique for displaying scalar field structure[21]. By their definition, isocontours represent the data only at discrete levels, and as such are an effective technique for determining the "shape" of objects in the scalar field. Shape extraction as defined by isocontours is well understood and appreciated in many applications, such as Medical Imaging, as isocontours in a density field may result in realistic models of skeletal structure, skin surface, or various organs[22]. Also implicit in their definition is the fact that isocontours are an incomplete representation of the scalar field, as one can only infer from an isocontour that the data to one side is above the isovalue, and the data to the other side is below the isovalue. With multiple isocontours, the scalar field effectively becomes segmented into a finite number of ranges, within which the structure remains unknown. The same claim of incompleteness can be made of any technique which only displays a portion of the field. Moreover it is not obvious which isovalues one should select and how namy of them [4].

Colormapping of scalar data defines a discrete or continuous range of colors onto which the scalar values are mapped. Use of color, though proven to be useful in many visualization techniques, introduces complications due of perceptual issues, such as colorblindness. Colormaps may also mislead the user, for example when small-scale structure in the data is washed out due to the large range of values taken on by the variable.

Scientific data which is time-varying in nature intensifies the problems with the methods described above. In the typical case, a scalar variable may take on a wide range of values over the course of a simulation, however at certain times during the simulation the range may be much smaller. With both isocontours and colormapped display, it is desirable to use the same isovalues and colormap for each time-step being displayed in order to reduce the possibility of introducing artifacts which may be misinterpreted as features. This requirement complicates the task of choosing a good colormap or selection of isovalues for a time-varying visualization.

In this paper we present a complementary scalar structure visualization technique which does not depend on the user to determine structure from the graphical display, but instead defines, computes, and displays the structure of a scalar field directly. Through detection of all critical points (saddles, maxima, and minima), we construct an embedded graph by computing integral curves in the gradient field from saddle points to an attached critical point, as illustrated in figure 1. Curves in
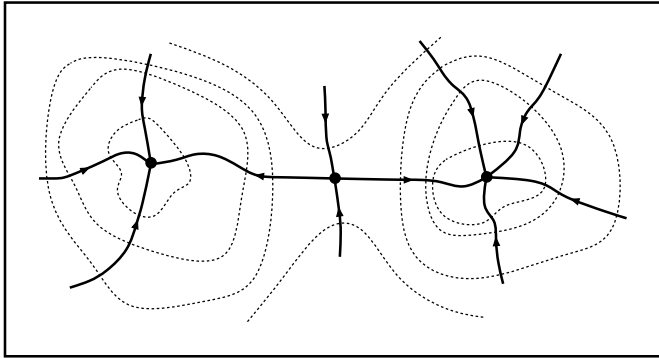
Figure 1: Isocontours (dotted) of part of a scalar field along with the critical points and integral curves

this topological graph are always perpendicular to isocontours of the scalar field[23], and we will demonstrate that these curves contain complementary information to that provided by display of isocontours or colormapped scalar fields, providing a method which is both useful in its own right and which also enhances the commonly used techniques for visualizing scalar fields. We further indicate that the definition of structure which is provided by the scalar topology proves useful in several additional visualization and image processing applications.

2 Related Work

Much of the work in enhancing colormapped visualization of scalar fields has dealt with determining "good" colormaps which effectively display the data. Bergman, et. al., define rules based on perception, user goals, and data characteristics to automatically select a colormap which will meet the user requirements[6]. Histogram equalization is a technique which spreads the data evenly over the range of colors, using the available color space to it's fullest[28]. The result is that each color in the colormap is used an equal number of times. Gershon[14] uses "Generalized Animation" to display otherwise static scalar data in a dynamic way, taking advantage of the ability of the visual system to detect dynamic changes. Animation draws attention to fuzzy details in the data which may not be detected in the static representation.

There has been several papers in detecting isocontours in 2d and 3d scalar data[21, 30]. Additional work concentrates on handling problems in regions containing saddle points which cause difficulty in determining the topological structure of the surface contained in the region [25, 31, 26]. The problem of detecting ridges and valleys in digital terrain has been treated in several papers[12]. McCormack, et. al. consider the problem of detecting drainage patterns in geographic terrain[24]. Interrante, et. al. have used ridge and valley detection on 3d surfaces to enhance the shape of transparently rendered surfaces[19]. Extrema graphs were used by Itoh and Koyamada to speed isocontour extraction[20]. A graph containing extreme points and boundary points of a scalar field can be guaranteed to intersect every isocontour at least once, allowing seed points to be generated by searching only the cells contained in the extrema graph.

Helman and Hesselink detect vector field topology by classifying the zeros of a vector field and performing particle tracing from saddle points[17]. The resulting partitioning consists of regions which are topologically equivalent to uniform flow. Globus, et. al. describe a software system for 3d vector topology and briefly note that the technique may also be applied to the gradient of a scalar field in order to identify maxima and minima[15]. Bader et. al. and Collard et. al. examine the gradient field of the charge density in a molecular system[2, 1, 10]. The topology of this scalar field represents the bonds linking together the atoms of the molecule. Bader goes on to show how features higher level structures in the topology represent chains, rings, an cages in the molecule. Bader's example is a defining motivation for developing the automatic extraction and visualization of topology from a scalar field. In many situations, topology provides a more intuitive and physically meaningful visualization. Grosse [16] also presents methods of approximating the scalar topology of the electron density function of proteins. One of his methods uses tensor product B-spline fits while the other scales Fourier coefficients of the electron density function.

3 Scalar Topology

Previous techniques for enhancing scalar field visualization attempt to address the inability of colormapping and isocontouring to capture and directly represent features in the data. We address this problem not through feature enhancement using existing visualization techniques, but through direct feature detection and display. For our purpose of detection and display, we define the topology of a scalar field S defined with domain D to consist of the following:

1. The local **maxima** of S

2. The local **minima** of S

3. The **saddle points** of S

4. Selected **integral curves** joining each of the above

Integral curves are defined as curves which are everywhere tangent to the gradient field of S. Intuitively, these curves represent the path followed by a heat-seeking particle in a temperature field, or the path followed by a ball rolling down a hill in a field of elevation values. In vector field topology, the curves advected in the flow field segment the field into regions which are topologically equivalent to uniform flow. In the case of scalar topology, integral curves segment the field into regions in which the gradient flow is uniform, or in other words, the scalar function is monotonic. Such a segmentation of the scalar field into regions of simple behavior reveals the structure of the scalar field for the visualization user.

We outline the procedure for visualization of scalar topology as follows:

1. Detect stationary (critical) points in S.

2. Classify stationary points.

3. Integrate selected integral curves in gradient field.

In the following subsections, we will define our model of a continuous scalar field and look at each of the steps defined above.

3.1 Scalar Field Model

In typical scientific applications, data is represented at the nodes of a mesh of elements and interpolated linearly across the interior of the elements. Such a data model is C^0 continuous and has a discontinuous gradient field, making it unsuitable for our purpose of tracing integral curves in the gradient field. We seek to construct a data model such that:

1. The original nodal data is interpolated.

2. The gradient at the boundaries is C^0 continuous.

3. Critical points in the scalar field are not removed, and the number introduced is kept small.
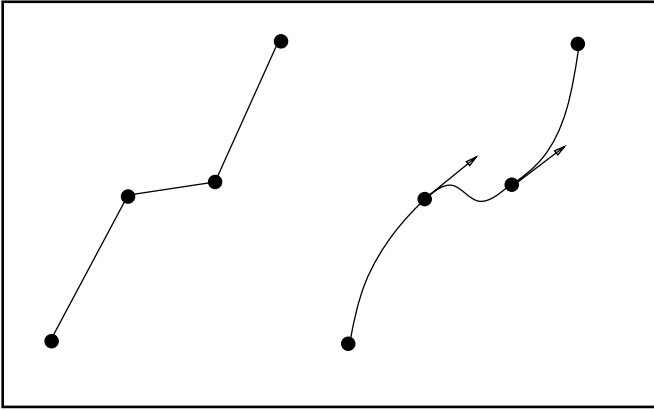

Figure 2: Artificial extreme points introduced by central differencing

We could satisfy the first two properties by computing derivatives by a method such as central differencing, which would uniquely define a C^1 continuous bi-cubic scalar interpolant [3]. However, such a choice of interpolant is likely to violate our third requirement by introducing critical points, as illustrated for the 1-D case in figure 2.

To address this problem, we use a "damped" central differencing scheme as described in the following sections. The resulting scalar field will remain a piecewise C^1 continuous bi-cubic function, which we represent in Bernstein-Bézier form as:

$$S = \sum_{i=0}^{3} \sum_{j=0}^{3} w_{i,j} B_i^3(x) B_j^3(y) \quad x, y \in [0, 1]$$

where

$$B_i^n(t) = \binom{n}{i} t^i (1 - t)^{n-i}$$

As a result, the derivatives of the scalar field can be represented as:

$$\frac{\partial S}{\partial x} = \sum_{i=0}^{2} \sum_{j=0}^{3} 3(w_{i+1,j} - w_{i,j}) B_i^2(x) B_j^3(y)$$

$$\frac{\partial S}{\partial y} = \sum_{i=0}^{3} \sum_{j=0}^{2} 3(w_{i,j+1} - w_{i,j}) B_i^3(x) B_j^2(y)$$

$$\frac{\partial^2 S}{\partial x \partial y} =$$

$$\sum_{i=0}^{2} \sum_{j=0}^{2} 9(w_{i+1,j+1} - w_{i,j+1} - w_{i,j} + w_{i+1,j}) B_i^2(x) B_j^2(y)$$

Having computed the damped partial derivatives and mixed partials for each vertex, the weights $w_{i,j}$ are computed according to the above equations. For $(x, y) = (0, 0)$, we get:

$$w_{0,0} = S_{0,0}$$

$$w_{0,1} = S_{0,0} + \frac{1}{3} \frac{\partial S}{\partial x}$$

$$w_{1,0} = S_{0,0} + \frac{1}{3} \frac{\partial S}{\partial y}$$

$$w_{1,1} = S_{0,0} + \frac{1}{3} \frac{\partial S}{\partial x} + \frac{1}{3} \frac{\partial S}{\partial y} + \frac{1}{9} \frac{\partial^2 S}{\partial x \partial y}$$

Similar equations follow for the other three vertices of a cell in 2D. The method presented for computing damped central differences in the following sections is based on the above equations for the weights of the surface S, and is developed with the goal of satisfying our scalar model criteria defined above.

For further information on smooth surface representations and for modeling scalar fields, see for example [18, 3]. There is also a sparse body of literature concerning curve and surface interpolation which retains shape, where shape is generally thought of in terms of monotonicity or convexity. [8, 9, 5, 13, 11].

3.1.1 One-dimensional derivatives

Consider the one-dimensional case of three points along a line, as pictured in figure 3. We compute the derivative at x_1 as follows:

- If $y_1 > y_0$ and $y_1 > y_2$, assign $\partial S/\partial x = 0$. Thus we preserve that a maximum in the linear field remains a critical point in the interpolated field.

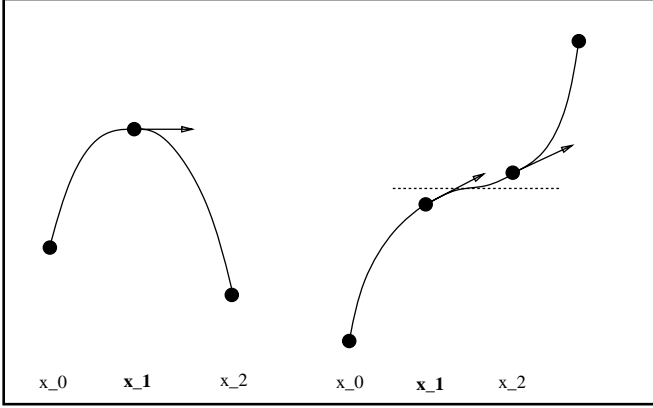- Likewise, if $y_1 < y_0$ and $y_1 < y_2$, assign $\partial S/\partial x = 0$.

Figure 3: Damped central differences maintain critical points

- Otherwise, the point data at x_0, x_1, and x_2 are monotonic, and we dampen the central difference as follows:

$$\partial S / \partial x = sign(x_1) min(abs(\frac{y_2 - y_0}{x_2 - x_0}),$$

$$abs(\frac{3(y_1 - y_0)}{2(x_1 - x_0)}), abs(\frac{3(y_2 - y_1)}{2(x_2 - x_1)}))$$

where $sign(x_1) = -1$ if the data is monotonically decreasing at x_1 and $sign(x_1) = 1$ if the data is monotonically increasing.

The first two conditions guarantee that extreme points of the linear line segments remain critical points in the cubic interpolant. The third condition is motivated by the control polygon of the resulting cubic curve. As illustrated in figure 3, damping the central difference by a multiple of the one sided difference guarantees that the control points between x_1 and x_2 will lie within the ranges $[y_1, (y_1 + y_2)/2]$ and $[(y_1 + y_2)/2, y_2]$, respectively. Thus, we assure that the control points within each segment will be monotonic, and guarantee that the derivative in this segment will not vanish, as illustrated in the closeup in figure 4
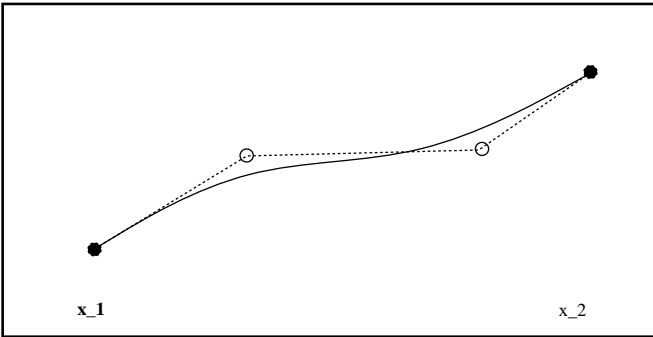

Figure 4: Closeup of segment from figure 2 illustrating guarantee of monotonicity

3.1.2 Two and higher dimensional derivatives

In two dimensions, the first partials $\partial S / \partial x$ and $\partial S / \partial y$ are handled as in the one-dimensional case, with one minor exception. Rather than scaling each component of the gradient, the central difference is taken in both directions, and the result is damped by the minimum of the scaling factors in either direction. In other words, in higher dimensions we dampen not each component of the gradient, but the magnitude of the gradient, leaving the direction the same as that computed from central differences. This simple extension of the one-dimensional case is sufficient to guarantee that critical points are not introduced along the edges in two dimensions, edges and faces in three dimensions, and so on.
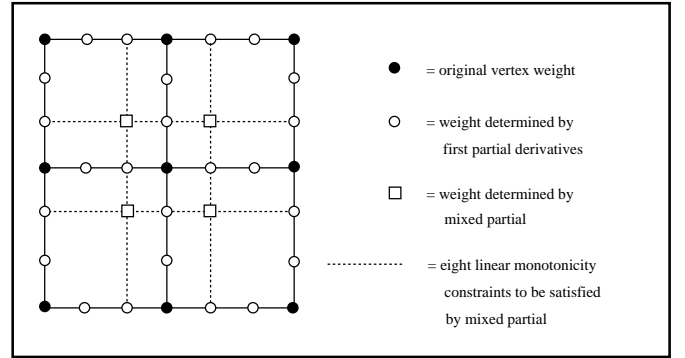

Figure 5: Constraints on the mixed partial derivative for 2D

What remains in the 2D case is to compute the mixed partial $\partial^2 S / \partial x \partial y$. For this, we again resort to the equations for computing the weights $w_{i,j}$. Having computed the first partials, our weights are fixed along all edges of the mesh, as illustrated in figure 5. We would like to constrain the mixed partial at each vertex such that the four interior weights adjacent to the vertex are guaranteed to satisfy the monotonicity condition in both directions, which is effectively equivalent to eight one-dimensional constraints. This is clearly overconstrained, and examples for which $\partial^2 S / \partial x \partial y$ cannot meet all constraints are easy to construct. We compute the eight linear constraints and examine them to see if there exists a simultaneous solution. If there is not, then we set $\partial^2 S / \partial x \partial y = 0$ in order to minimize the twist on the resulting patch[11]. We point out the fact that we maintain monotonicity along the edges to guarantee that a bi-linear cell which contains a saddle point will contain a saddle in the shape preserving interpolated field.

Similar sets of linear constraint equations are examined and resolved for the higher order partials in 3D and higher, as illustrated in figure 6.

3.2 Computing Critical Points

Critical points of a scalar function are defined as points at which the gradient vanishes[23]. For a bicubic function (2D) or tricubic function (3D), computing the positions of critical points amounts to solving a non-linear system of equations. How-
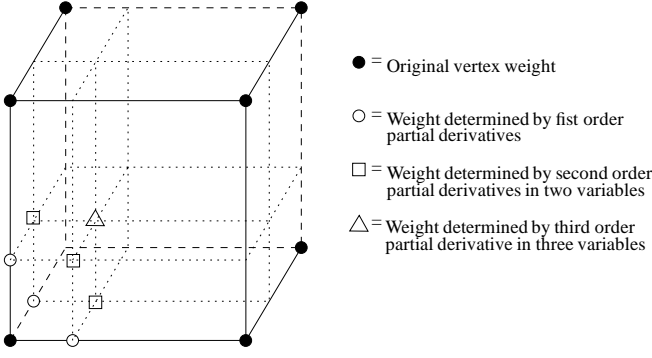
Figure 6: Constraints on the mixed partial derivatives for 3D

Figure 7: Some of the scalar critical points

ever, due to the special construction of our interpolant, we have knowledge about where the critical points will occur, and can compute them quite efficiently.

Critical points which occur at the vertices of the mesh will be preserved, and can be computed from the bilinear or trilinear field respectively, with the guarantee that they exist as well in the higher order shape preserving interpolant. Critical points interior to a cell will occur in locations at which the monotonicity constraint could not be met. In smooth parts of the field, there will be no problem computing a monotone field, which will guarantee the absence of critical points. In cells at which constraints were violated, we perform subdivision of the cell in order to locate the critical points, followed by Newton-Rhapson iteration to refine the positions of the zeroes. Saddles from the initial bilinear or trilinear mesh can be approximated by computing the position of the bilinear or trilinear saddle analytically, followed by iteration in the bi-cubic or tri-cubic field, respectively.

3.3 Classification of Critical Points

Qualitative information about the behavior of the gradient field near a critical point is obtained by analysis of the Hessian of S, given for 2D:

$$\begin{bmatrix} \frac{\partial^2 S}{\partial x^2} & \frac{\partial^2 S}{\partial x \partial y} \\ \frac{\partial^2 S}{\partial y \partial x} & \frac{\partial^2 S}{\partial y^2} \end{bmatrix}$$

The eigenvalues and eigenvectors of the above matrix determine the behavior of the gradient field and hence the scalar field near the critical point, much the same as for the behavior of a general vector field[7, 17]. One difference to note is that for a gradient field, the matrix of derivatives is symmetric ($\partial^2 S/\partial x \partial y = \partial^2 S/\partial y \partial x$), and therefore the eigenvalues will all be real. This is intuitively expected, as imaginary eigenvalues indicate rotation about the critical point, and a gradient field is an irrotational vector field. This observation allows us to simplify the classification of critical points as depicted in figure 7.

A positive eigenvalue corresponds to gradient flow away from the critical point, while a negative eigenvalue indicates
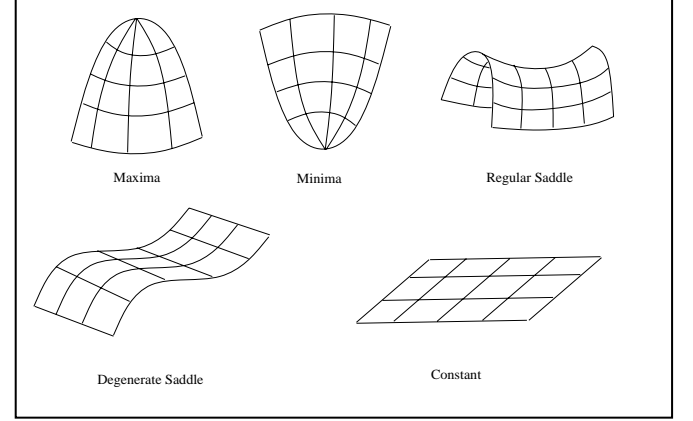
gradient flow toward the critical point. In the case of a saddle point, there is gradient flow toward and away from the critical point, distinguishing it from the field behavior near other critical points. In this case, the eigenvectors corresponding to the positive and negative eigenvalues define the principal directions of the flow toward and away from the saddle, respectively. It is this property that will be used in the next section to compute critical curves in the gradient field.

3.4 Tracing Integral Curves

Having computed and classified the critical points, the final step for computing the scalar topology is the tracing of selected critical curves between the detected points. Even for three and higher dimensional scalar fields we restrict our focus to only computing critical curves, and ignore critical surfaces and hypersurfaces and other degeneracies in the field,

Saddle points have the property that the eigenvectors of the Hessian are the separatrices of the saddle. A particle following the gradient field along the these directions will come to rest at the saddle point, while particles slightly to either side of the separatrices will diverge rapidly near the point. It is for this reason that saddle points and the critical curves associated with their separatrices are useful in determining the structure of a scalar field. The number of critical curves emanating from saddles along separatrices is twice the field dimension. In 2D, four critical curves are computed for each saddle point, two in the direction corresponding to the positive eigenvalue, and two in the direction corresponding to the negative eigenvalue. In 3D, the number is six, and so on.

Integral curves are computed using the following 4th order adaptive step Runge Kutta integration in the gradient field[27], where Δt is the time step which adapts per iteration, and $\vec{x}_n$ is a field point :

1. $\vec{k}_1 = \Delta t \vec{v}(\vec{x}_n)$

2. $\vec{k}_2 = \Delta t \vec{v}(\vec{x}_n + \frac{\vec{k}_1}{2})$

3. $\vec{k}_3 = \Delta t \vec{v}(\vec{x}_n + \frac{\vec{k}_2}{2})$

4. $\vec{k}_4 = \Delta t \vec{v}(\vec{x}_n + \vec{k}_3)$

5. $\vec{x}_{n+1} = \vec{x}_n + \frac{\vec{k}_1}{6} + \frac{\vec{k}_2}{3} + \frac{\vec{k}_3}{3} + \frac{\vec{k}_4}{6} + O(\Delta t^5)$

The initial position for the iterative stepping is placed a small distance from the saddle point along the appropriate eigenvector. The steps are bounded such that we take no less than 5 steps per cell, maintaining a high level of accuracy. Computation of the critical curve ends when we reach the vicinity of another critical point within a certain ϵ, in which case the curve terminates at that point. Other curves may end at the boundaries of the mesh.

4 Quality Comparison

Here we compare the qualities of scalar topology visualization with those of isocontours and colormapping.

Integral curves are everywhere orthogonal to isocontours. The two techniques arise from an orthogonal definition of "structure" for a scalar variable. Contours are an attempt to compute and display the exact shape of an object in a scalar field, while the topology graph attempts to show the relations among all such objects in the field, without giving the details of shapes of particular objects. Note that scalar field topology is invariant under translation and uniform scaling. This quality is very similar to colormapping of scalar variables, in which the entire range of variables is mapped into a color space. Translation and scaling of the scalar variables changes only the mapping function, not the result.

5 Examples

Figures 8, 9, 10 and the top two pictures of the Color Plate, demonstrate the use of scalar topology along with both isocontours and colormapped visualizations of density in an off-axis pion collision. Figure 8 uses a simple greyscale colormap, and it is clear that much of the area of interest in the center is washed out. Figure 9 uses a hue-based colormap and adds isocontours of three isovalues to reveal more of the structure and aid the perception. In figure 10, we show the scalar topology of density. This image clearly brings out the detail of the structure of the variable. The top figures of the Color Plate show a closeup of the interesting topological regions, as well as shows a combination of all three visualization techniques.

While small scale structure is important in many scientific applications, in some circumstances the visualization user is interested only in large scale structure. For this situation, we apply a filter to smooth the data before applying the topology detection algorithm. Figures 11, 12 show two visualizations of topology in a scalar field representing wind speed. In figure 11, the unfiltered scalar field topology reveals some noise in the data. Figure 12 shows the topology for the same data after a Gaussian filter has been applied.

The middle figures of the color plate shows an example of scalar topology applied to a mathematically defined surface. In the left figure the scalar topology is displayed. In the right figure both topology and four isocontours are displayed. Notice that even with four isolevels displayed, there are critical points within contour regions which are not revealed like the two maxima on the bottom left that are not separated by any isocontour.

The bottom figures of the color plate show an example of scalar topology applied to a 3D scalar fields (the wave function computed for a high potential iron protein).

6 Other Applications

Computation of scalar topology has the potential to serve many other visualization and image processing applications. We mention only a few here:

Data Correlation - Due in part to the invariance under translation and scaling, scalar topology is useful in visually determining linear correlation between multiple scalar variables.

Image Co-registration - Scalar topology in adjacent planes provides a "1D skeleton" which may be used to align the planes.

Warping/Morphing - Editing of the scalar backbone may be used to apply a warping effect to an image, or to warp between the backbones of two similar images.

Mesh Reduction - The scalar topology may serve as a guide to aid in computation of reduced resolution meshes.

Surface Triangulation - Adaptive triangulation of arbitrary mathematical surfaces by decomposition into monotonic patches which may be subdivided to an arbitrary precision.

7 Conclusions

Existing scalar visualization techniques lack the ability to explicitly present the structure of a scalar field to the user. We have presented a definition of scalar structure and a straightforward algorithm for computing and displaying the structure. For typical scientific data, the scalar data model remains true to the original linear data, minimizing introduction of false critical points, and also simplifying the detection of critical points.

The resulting topology visualization serves to both provide information which is not available in commonly used scalar visualization techniques, as well as reinforcing or enhancing the information provided by common visualization techniques. Furthermore, computation of scalar topology offers promise toward improving several visualization and image processing applications.

References

[1] R. Bader. *Atoms in Molecules*. Clarendon Press, Oxford, 1990.

[2] R. Bader, T. Tung Nguyen-Dang, and Y. Tal. Quantum topology of molecular charge distributions ii. molecular structure and its charge. *Journal of Chemical Physics*, 70:4316–4329, 1979.

[3] C. Bajaj. Modelling physical fields for interrogative visualization. In Tim Goodman and Ralph Martin, editor, *The Mathematics of Surfaces VII*, pages 241–262. Information Geometers Ltd., 1997.

[4] C. L. Bajaj, V. Pascucci, and D. R. Schikore. The contour spectrum. In R. Yagel and H. Hagen, editors, *Proceeding Visualization '97*, pages 167–173, Phoenix, AZ, October 1997. IEEE Computer Society & ACM SIGGRAPH.

[5] R. Beatson and Z. Ziegler. Monotonicity preserving surface interpolation. In *SIAM J. Numer. Anal.*, volume 22, pages 401–411, April 1985.

[6] L. Bergman, B. Rogowitz, and L. Treinish. A rule-based tool for assisting colormap selection. In G. M. Nielson and D. Silver, editors, *Visualization '95 Proceedings*, pages 118–125, October 1995.

[7] W. Boyce and R. DiPrima. *Elementary Differential Equations and Boundary Value Problems*. John Wiley and Sons, Inc., New York, fifth edition, 1992.

[8] R. Carlson and F. Fritsch. Monotone piecewise bicubic interpolation. In *SIAM J. Numer. Anal.*, volume 22, pages 386–400, April 1985.

[9] R. Carlson and F. Fritsch. An algorithm for monotone piecewise bicubic interpolation. In *SIAM J. Numer. Anal.*, volume 26, pages 230–238, February 1989.

[10] K. Collard and G. Hall. Orthogonal trajectories of the electron density. *International Journal of Quantum Chemistry XII*, 0:623–637, 1977.

[11] S. Dodd, D. McAllister, and J. Roulier. Shape-preserving spline interpolation for specifying bivariate functions on grids. In *IEEE Computer Graphics and Applications*, pages 70–79, September 1983.

[12] R. J. Fowler and J. J. Little. Automatic extraction of irregular network digital terrain models. In *Computer Graphics (SIGGRAPH '79 Proceedings)*, volume 13(3), pages 199–207, August 1979.

[13] F. Fritsch and R. Carlson. Monotone piecewise cubic interpolation. In *SIAM J. Numer. Anal.*, volume 17, pages 238–246, April 1980.

[14] N. D. Gershon. Visualization of fuzzy data using generalized animation. In A. E. Kaufman and G. M. Nielson, editors, *Visualization '92 Proceedings*, pages 268–273, October 1992.

[15] A. Globus, C. Levit, and T. Lasinski. A tool for visualizing the topology of three-dimensional vector fields. In G. M. Nielson and L. Rosenblum, editors, *Visualization '91 Proceedings*, pages 33–40, October 1991.

[16] E. Grosse. *Approximation and Optimization of Electron Density Maps*. PhD thesis, Stanford University, 1980.

[17] J. Helman and L. Hesselink. Visualizing vector field topology in fluid flows. *IEEE Computer Graphics and Applications*, 11(3), 1991.

[18] J. Hoscheck and D. Lasser. *Fundamentals of Computer Aided Geometric Design*. A K Peters, Wellesley, Massachusetts, 1993.

[19] V. Interrante, H. Fuchs, and S. Pizer. Enhancing transparent skin surfaces with ridge and valley lines. In G. M. Nielson and D. Silver, editors, *Visualization '95 Proceedings*, pages 52–59, October 1995.

[20] T. Itoh and K. Koyamada. Isosurface extraction by using extrema graphs. In R. D. Bergeron and A. E. Kaufman, editors, *Visualization '94 Proceedings*, pages 77–83, October 1994.

[21] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In Maureen C. Stone, editor, *Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21, pages 163–169, July 1987.

[22] W. Lorenson. Marching through the visible man. In G. M. Nielson and D. Silver, editors, *Visualization '95 Proceedings*, pages 368–373, October 1995.

[23] J. Marsden and A. Tromba. *Vector Calculus*. W. H. Freeman and Company, New York, third edition, 1988.

[24] J. E. McCormack, M. N. Gahegan, S. A. Roberts, J. Hogg, and B. S. Hoyle. Feature-based derivation of drainage networks. *Int. Journal of Geographical Information Systems*, 7(3):263–279, 1993.

[25] B. K. Natarajan. On generating topologically consistent isosurfaces from uniform samples. Technical Report HPL-91-76, Hewlett-Packard, June 1991.

[26] G. M. Nielson and B. Hamaan. The asymptotic decider: Resolving the ambiguity of marching cubes. In G. M. Nielson and L. Rosenblum, editors, *Visualization '91 Proceedings*, pages 83–91, October 1991.

Figure 8: Visualization of density in a pion collision simulation: Greyscale colormapping
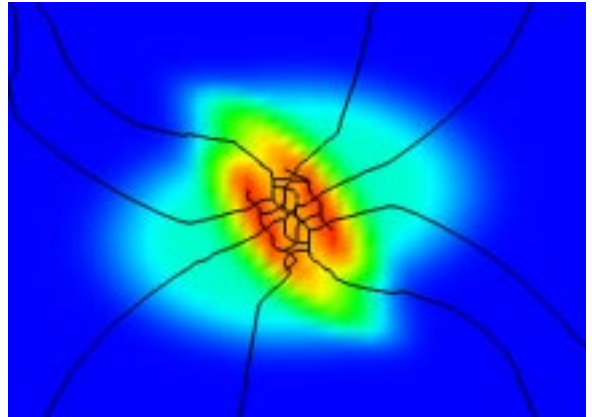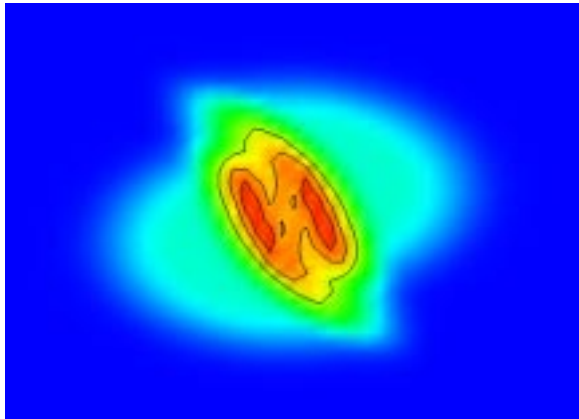

Figure 9: Visualization of density in a pion collision simulation: Isocontours overlayed on hue based colormapping

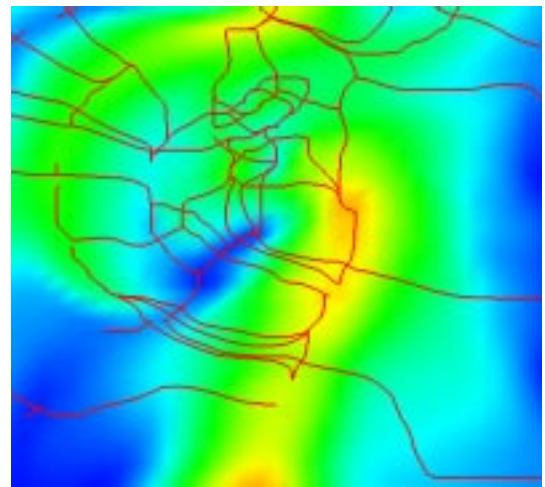Figure 10: Visualization of density in a pion collision simulation: Topology overlayed on hue based colormapping

Figure 11: Visualization of wind speed from a climate model: Scalar topology for noisy data
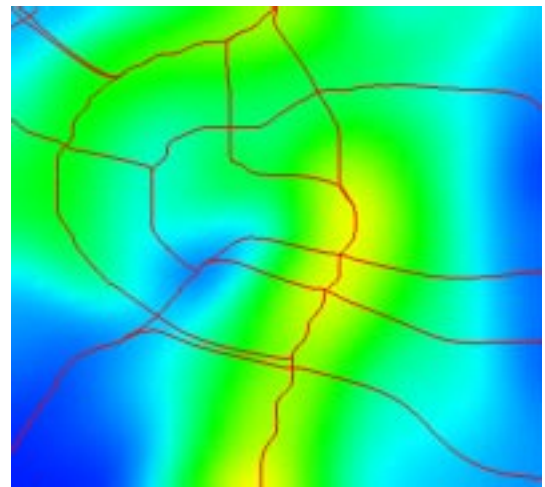

Figure 12: Visualization of wind speed from a climate model: After applying a Gaussian filter

[27] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C, Second Edition*. Cambridge University Press, 1992.

[28] A. Rosenfeld and A. Kak. *Digital Picture Processing*. Academic Press, San Diego, 1982.

[29] E. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 1983.

[30] Jane Wilhelms and Allen Van Gelder. Octrees for faster isosurface generation extended abstract. In *Computer Graphics (San Diego Workshop on Volume Visualization)*, volume 24, pages 57–62, November 1990.

[31] Jane Wilhelms and Allen Van Gelder. Topological considerations in isosurface generation extended abstract. In *Computer Graphics (San Diego Workshop on Volume Visualization)*, volume 24, pages 79–86, November 1990.